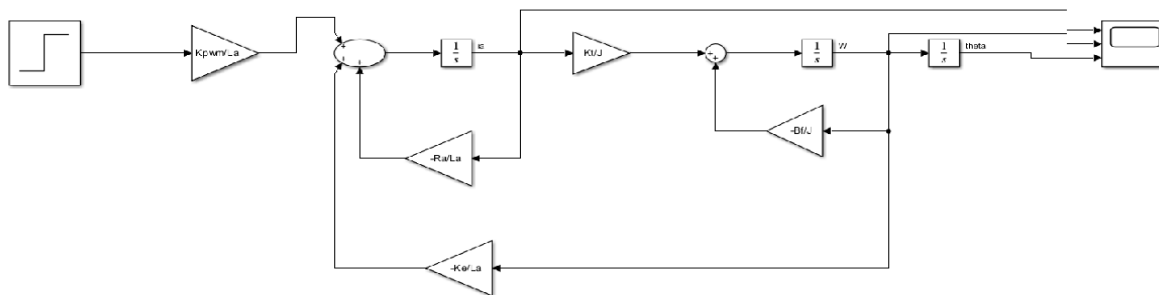
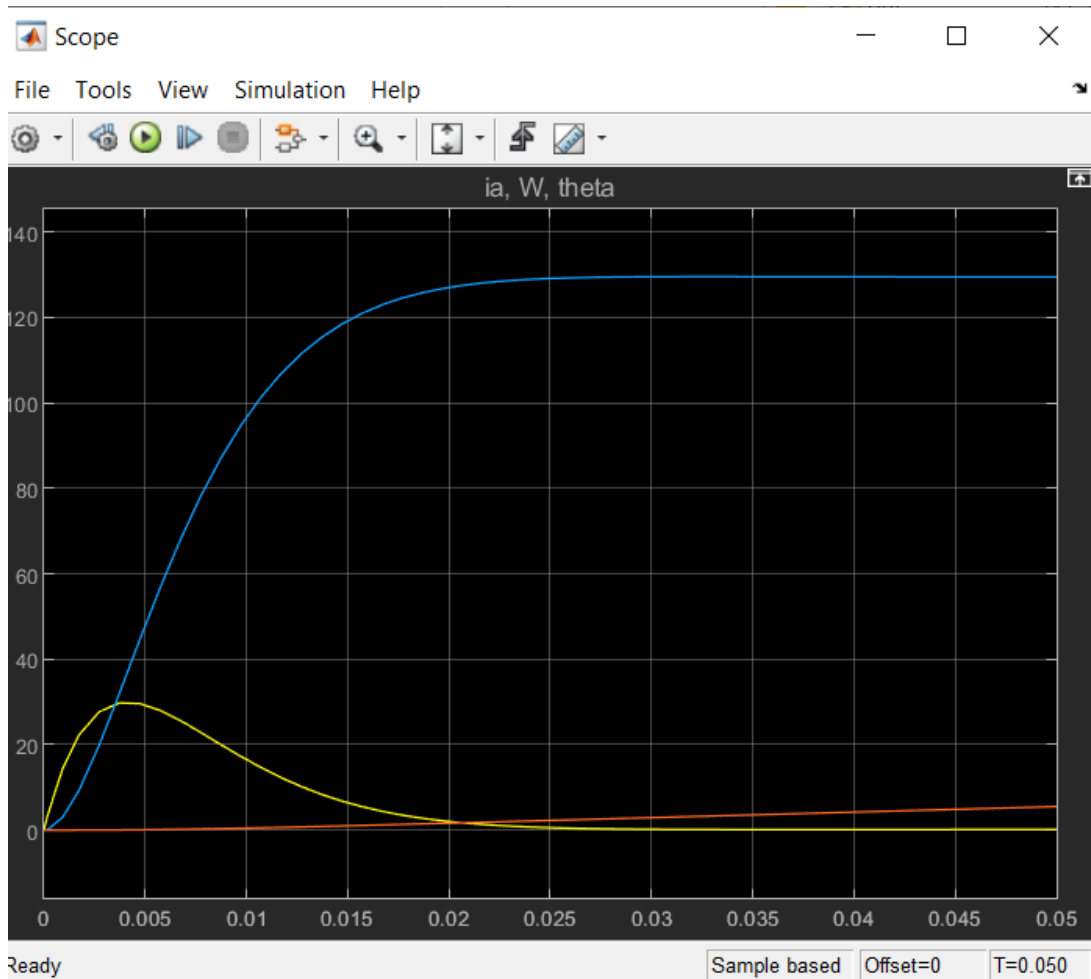


LAB 13 - POLE PLACEMENT

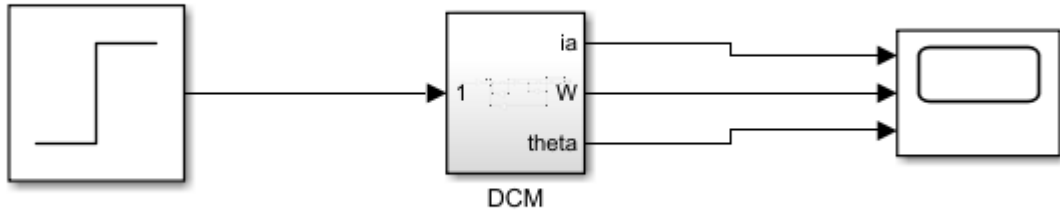
```
clear all
% pentru schema din simulink avem urmatorii parametri de model
Ra=0.92;
La=2e-3;
Ke=0.2960;
Kt=0.2939;
Bf=3.35e-4;
J=7.01e-4;
Kpwm=38.4615;
% Schema motorului in simulink
```



```
% Outputul pe scope
```



```
% Rotorul(portocaliu) creste la infinit  
% Angular velocity(galben), initial forteaza motorul sa porneasca, dupa  
% care revine la 0 deoarece motorul a ajuns in regim stationar.  
% Turatia(albastru)  
  
% Se selecteaza tot ce este intre step si scope, transformandu-l intru-un  
% subsistem
```

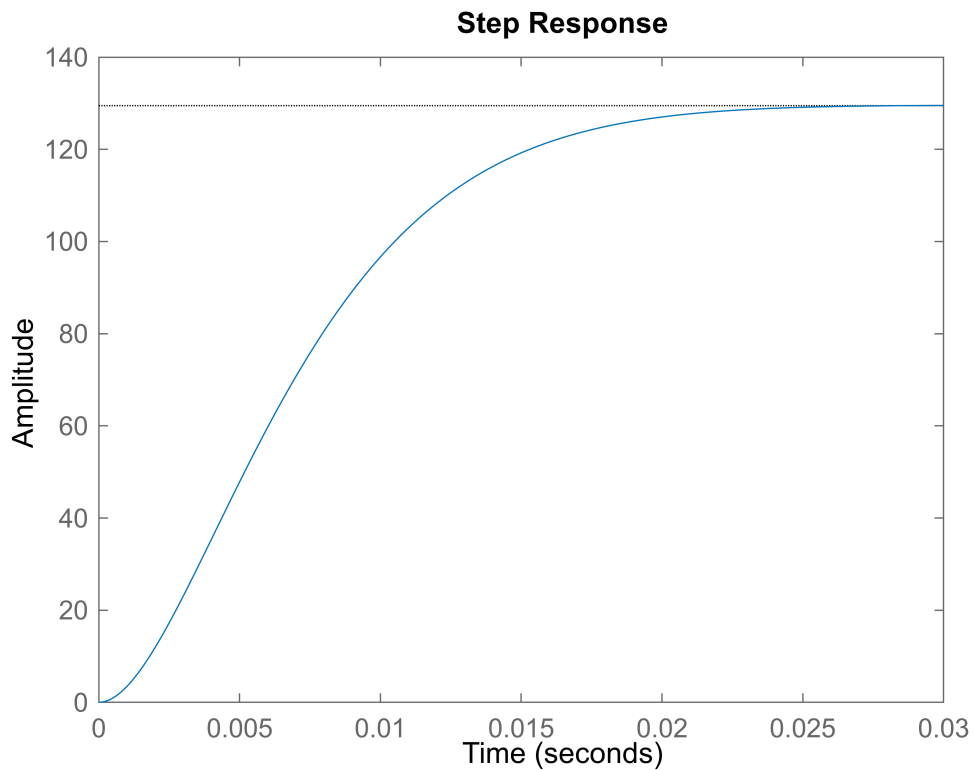


% Se doreste controlul turatiei, asa ca in primul rand se defineste state
 % space model-ul care va avea urmatoarea forma:

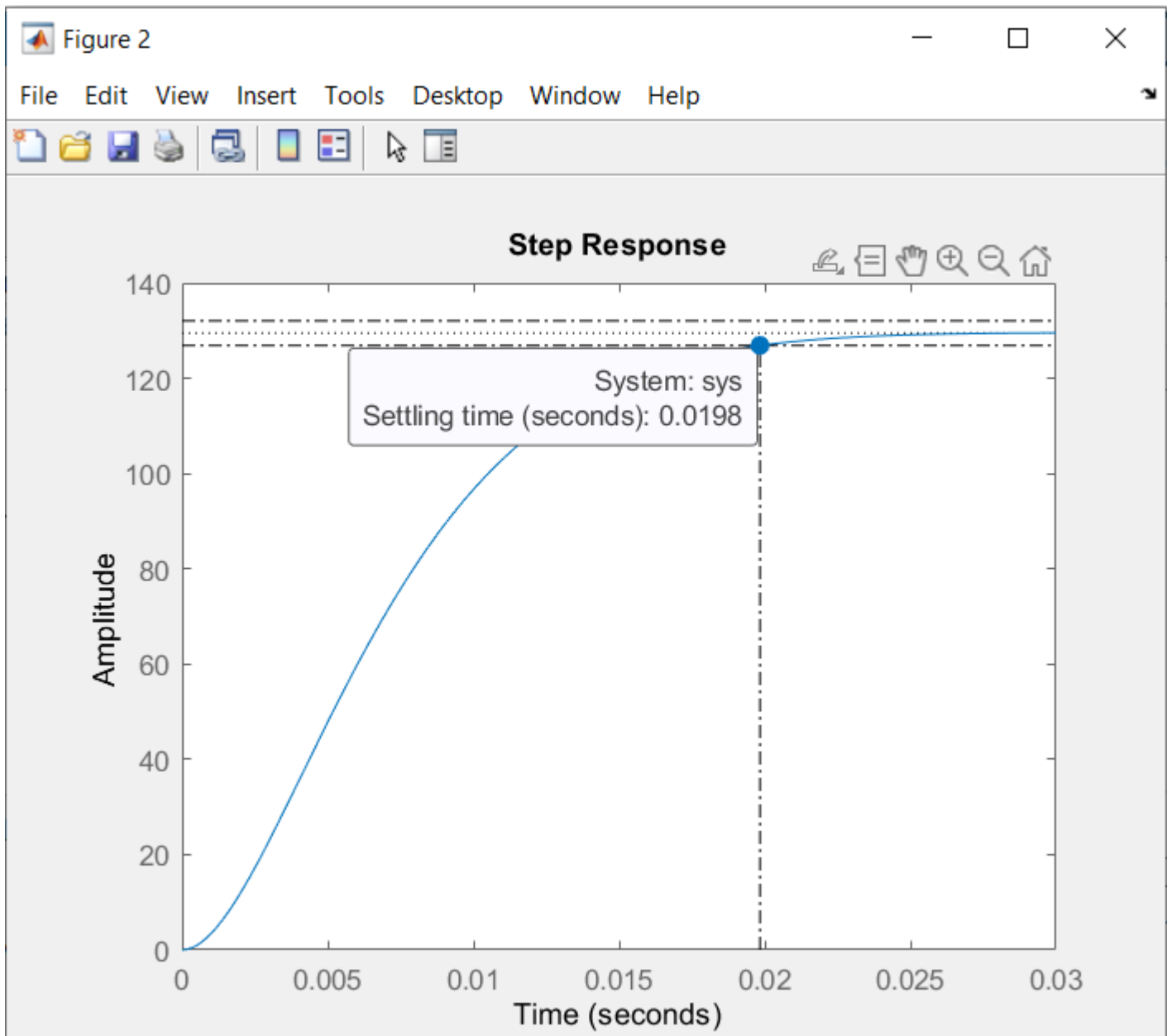
```
A=[-Ra/La, -Ke/La; Kt/J,-Bf/J];
B=[Kpwm/La,0;0 -1/J];
C=[0,1];
D=[0,0];
```

% Ne dorim sa reducem settling time-ul

```
step(A,B,C,D,1);
```



% Se observa ca settling time-ul e de 0.0198



% Putem reduce ts cu 10%, folosind plasarea polilor
`eig(A)`

```
ans = 2x1 complex
102 ×
-2.3024 + 0.9623i
-2.3024 - 0.9623i
```

`p=eig(A)+eig(A)/10`

```
p = 2x1 complex
102 ×
-2.5326 + 1.0585i
-2.5326 - 1.0585i
```

% Inainte sa calculam reactia starii, trebuie verificata matricea de

```
% controlabilitate
```

```
Co=ctrb(A,B(:,1));  
R=rank(Co)
```

```
R = 2
```

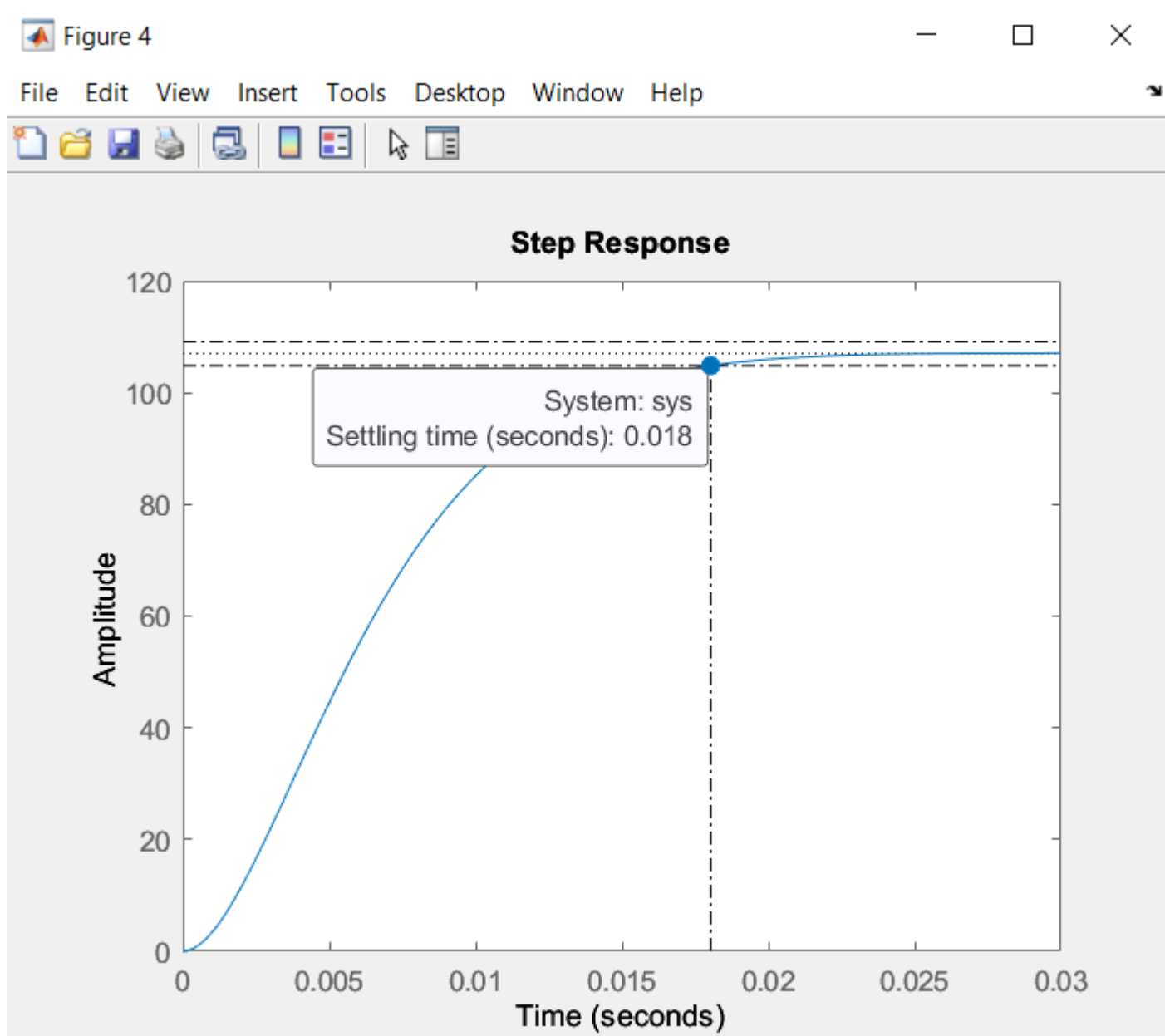
```
% Se observa ca rankul matricii Co este egal cu dimensiunea sa, deci  
% sistemul este controlabil ceea ce rezulta ca putem aplica metoda plasarii  
% polilor
```

```
Kx=acker(A,B(:,1),p)
```

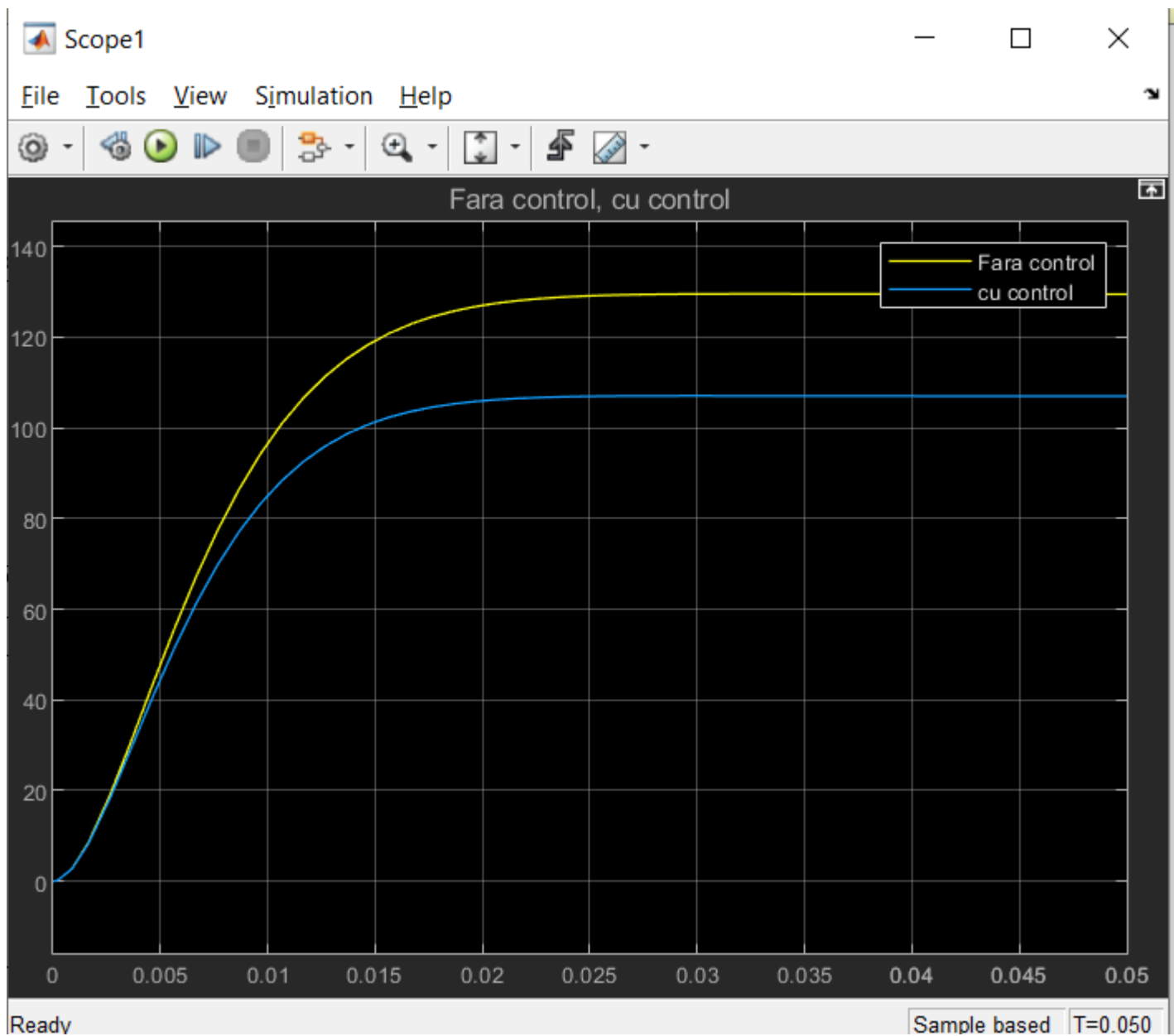
```
Kx = 1×2  
    0.0024    0.0016
```

```
Ar=A-B(:,1)*Kx;  
Br=B;  
Cr=C;  
Dr=D;
```

```
% Se observa ca settling time-ul a scazut
```



```
% Cand implementam partea de control in simulink se observa ca  
% se schimba valoarea stationara, asa ca trebuie sa le aducem  
% la aceeasi valoare
```



% Pentru a le aduce la aceeasi valoare stationara

```
Fx_ol=inv((C*inv(-A)*B(:,1)));  
Fx_cl=inv((C*inv(-Ar)*B(:,1)));
```

